Name: \_\_\_Joshua Schmidt\_\_\_\_\_\_ Date: \_\_\_\_10/10/19\_\_\_\_\_\_

Pledge: *\_\_\_\_\_I pledge my honor that I have abided by the Stevens Honor System.\_\_\_*

Use the Master Theorem to find the complexity of each recurrence relation listed below.

1. a = 1, b = 2, d = 2, 1 < 22

Complexity: \_\_\_

a = 4, b = 2, d = 2, 4 = 22  
Complexity: \_\_\_\_\_

a = 3, b = 3, d = 0.5, 3 > 3.5,   
Complexity: \_\_\_\_

For each function below, write the recurrence relation for its running time and then use the Master Theorem to find its complexity.

1. **int** f(**int** arr[], **int** n) {

**if** (n == 0) {

**return** 0;

}

**int** sum = 0;  
 **for** (**int** j = 0; j < n; ++j) {  
 sum += arr[j];  
 }  
 **return** f(arr, n / 2) + sum + f(arr, n / 2);  
}  
  
Recurrence: \_\_\_\_\_\_

a = 2, b = 2, d = 1, 2 = 21

Complexity: \_\_\_\_\_\_

1. **void** g(**int** n, **int** arrA[], **int** arrB[]) {

**if** (n == 0) {

**return**;

}

**for** (**int** i = 0; i < n; ++i) {  
 **for** (**int** j = 0; j < n; ++j) {  
 arrB[j] += arrA[i];

}

}  
 g(n / 2, arrA, arrB);  
}

Recurrence: \_\_\_\_\_\_\_\_\_\_\_

a = 1, b =2, d = 2, 1<22

Complexity: \_\_\_\_\_\_